

Form PTO-1390		U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER P21102
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371		U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 09/856514	
INTERNATIONAL APPLICATION NO. PCT/SG99/00077	INTERNATIONAL FILING DATE 15 July 1999	PRIORITY DATE CLAIMED 16 December 1998	
TITLE OF INVENTION A METHOD OF TRANSFERRING AN ACTIVE APPLICATION FROM A SENDER TO A RECIPIENT			
APPLICANT(S) FOR DO/EO/US Hwee Hwa PANG and Bin GUO			
Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information.			
1. <input checked="" type="checkbox"/> This is a FIRST submission of items concerning a filing under 35 U.S.C. 371. 2. <input type="checkbox"/> This is a SECOND or SUBSEQUENT submission of items concerning a filing under 35 U.S.C. 371. 3. <input checked="" type="checkbox"/> This is an express request to promptly begin national examination procedures (35 U.S.C. 371(f)). 4. <input checked="" type="checkbox"/> The US has been elected by the expiration of 19 months from the priority date (PCT Article 31). 5. <input checked="" type="checkbox"/> A copy of the International Application as filed (35 U.S.C. 371(c)(2)) a. <input checked="" type="checkbox"/> is attached hereto (required only if not communicated by the International Bureau). b. <input type="checkbox"/> has been communicated by the International Bureau. c. <input type="checkbox"/> is not required, as the application was filed in the United States Receiving Office (RO/US). 6. <input type="checkbox"/> An English language translation of the International Application as filed (35 U.S.C. 371 (c)(2)). 7. <input type="checkbox"/> Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3)) a. <input type="checkbox"/> are attached hereto (required only if not communicated by the International Bureau). b. <input type="checkbox"/> have been communicated by the International Bureau. c. <input type="checkbox"/> have not been made; however, the time limit for making such amendments has NOT expired. d. <input type="checkbox"/> have not been made and will not be made. 8. <input type="checkbox"/> An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)) 9. <input checked="" type="checkbox"/> An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)). "Unexecuted" 10. <input type="checkbox"/> An English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (U.S.C. 371(c)(5)). Items 11 to 16 below concern other document(s) or information included: 11. Assignee: <u>KENT RIDGE DIGITAL LABS of SINGAPORE</u> 12. <input type="checkbox"/> An Information Disclosure Statement under 37 CFR 1.97 and 1.98. 13. <input type="checkbox"/> An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included. 14. <input checked="" type="checkbox"/> A FIRST preliminary amendment. <input type="checkbox"/> A SECOND or SUBSEQUENT preliminary amendment. 15. <input type="checkbox"/> A substitute specification. 16. <input type="checkbox"/> A change of power of attorney and/or address letter. 17. <input type="checkbox"/> Figure of Drawing to be published _____ 18. <input checked="" type="checkbox"/> Other items or information: Cover Sheet and International Application as published. PCT/ISA/210. Cover Letter under 35 USC 371 and 1.495 Claim of Priority.			

U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 09/856514		INTERNATIONAL APPLICATION NO. PCT/SG99/00077		ATTORNEY'S DOCKET NUMBER P21102	
19. <u> </u> The following fees are submitted: Basic National Fee (37 CFR 1.492(a)(1)-(5)): Search report has been prepared by the EPO or JPO. \$ 860.00 International preliminary examination fee paid to USPTO (37 CFR 1.482). \$ 690.00 No international preliminary examination fee paid to USPTO (37 CFR 1.482) but international search fee paid to USPTO(37 CFR 1.445(a)(2)). \$ 710.00 Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO. \$1,000.00 International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4). \$ 100.00 ENTER APPROPRIATE BASIC FEE AMOUNT =				CALCULATIONS	PTO USE ONLY
				\$860.00	
Surcharge of \$130.00 for furnishing the oath or declaration later than <u> </u> 20 <u> </u> 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				\$	
Claims	Number Filed	Number Extra	RATE		
Total Claims	19 - 20 =	0	X \$18.00	\$0.00	
Independent Claims	2 - 3 =	0	X \$80.00	\$0.00	
Multiple dependent claim(s) (if applicable)			+ \$270.00	\$0.00	
TOTAL OF ABOVE CALCULATIONS =				\$860.00	
<u> </u> Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by <u>1/2</u> .				\$	
SUBTOTAL =				\$860.00	
Processing fee of \$130.00 for furnishing the English translation later than <u> </u> 20 <u> </u> 30 months from the earliest claimed priority date (37 CFR 1.492(f)).				+	
Extension of Time fee in the amount of \$					
TOTAL NATIONAL FEE =				\$860.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). \$40.00 per property				+	
TOTAL FEES ENCLOSED =				\$860.00	
				Amount to be refunded	\$
				Charged	\$
a. <u> X </u> A check in the amount of <u>\$860.00</u> to cover the above fees is enclosed. b. <u> </u> Please charge my Deposit Account No. <u> </u> in the amount of \$ <u> </u> to cover the above fees. c. <u> X </u> The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. <u>19-0089</u> . NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status. SEND ALL CORRESPONDENCE TO CUSTOMER NO. 7055 AT THE PRESENT ADDRESS OF: Bruce H. Bernstein GREENBLUM & BERNSTEIN, P.L.C. 1941 Roland Clarke Place Reston, VA 20191 (703) 716-1191					
				Signature <u>Leslie H. Papernon Reg. 16</u> SIGNATURE Bruce H. Bernstein <u>33,329</u> NAME 29.027 REGISTRATION NUMBER	

P21102.A01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Hwee Hwa PANG et al.

Serial No : Not Yet Assigned (National Stage of PCT/SG99/00077)

Filed : Concurrently Herewith (International Filing Date July 15, 1999)

For : A METHOD OF TRANSFERRING AN ACTIVE APPLICATION FROM
A SENDER TO A RECIPIENT

PRELIMINARY AMENDMENT

Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

Prior to calculation of the filing fees and the examination of the above-identified patent application on the merits, the Examiner is respectfully requested to amend the claims as follows:

IN THE CLAIMS

Please amend the claims as follows (a marked-up copy of the claim amendments is provided as an attachment to this Amendment):

9. (Amended-Clean Text) A method as claimed in claim 1 wherein prior to step (c) the hibernaculum is transferred by the recipient to a further location and opened from the further location.

P21102.A01

11. (Amended-Clean Text) A method as claimed in claim 6 wherein, prior to step (c) the reference is transferred by the recipient to a further mailbox or personal information manager and opened from there.

15. (Amended-Clean Text) A method as claimed in claim 1 wherein the sender and recipient are different machines.

16. (Amended-Clean Text) A method as claimed in claim 1 wherein the sender and recipient are different users.

17. (Amended-Clean Text) A method as claimed in claim 1 wherein the sender and recipient are the same, the steps of sending and recovering being spaced in time.

18. (Amended-Clean Text) Apparatus for performing the method of claim 1.

Please add new claim 19 as follows:

---19. Apparatus for performing the method of claim 12.---


P21102.A01

REMARKS

By the above amendment, claims 9, 11, 25, 16, 17, and 18 have been amended and claim 19 has been added to delete multiple dependency.

If there should be any questions, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,
Hwee Hwa PANG et al.


Bruce H. Bernstein
Reg. No. 29,027

Reg. No. 33,329

June 12, 2001
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191

MARKED-UP COPY OF AMENDED CLAIMS

9. (Amended) A method as claimed in claim 1 [any one of the preceding claims] wherein prior to step (c) the hibernaculum is transferred by the recipient to a further location and opened from the further location.

11. (Amended) A method as claimed in claim 6 [any one of claims 6 to 8] wherein, prior to step (c) the reference is transferred by the recipient to a further mailbox or personal information manager and opened from there.

15. (Amended) A method as claimed in claim 1 [any one of the preceding claims] wherein the sender and recipient are different machines.

16. (Amended) A method as claimed in claim 1 [any one of the preceding claims] wherein the sender and recipient are different users.

17. (Amended) A method as claimed in claim 1 [any one of claims 1 to 14] wherein the sender and recipient are the same, the steps of sending and recovering being spaced in time.

18. (Amended) Apparatus for performing the method of claim 1 [any one of the preceding claims].

A METHOD OF TRANSFERRING AN ACTIVE APPLICATION FROM A SENDER
TO A RECIPIENT

BACKGROUND AND FIELD OF THE INVENTION

5

This invention relates to a method of transferring an active application from a sender to a recipient.

Many user tasks performed on a computer span across multiple log-in sessions. However, the computing process, that is to say the combination of data, program module(s) and current execution state, that implements a user task typically does not live beyond a log-in session. This forces the user to save out the data, from which a new computing process will be constructed later to attempt to resume the user task, possibly on another machine, and possibly for another user. As a result, an application program either has to be written to save out all pertinent data, which is non-trivial, or, more often than not, the user has to put up with losing certain states of the original computing process. An example is Web surfing, where the navigation history is lost even though individual Web pages can be saved.

It is an object of the invention to address this problem.

SUMMARY OF THE INVENTION

According to the invention in a first aspect, there is provided a method of transferring an active application from a sender to a recipient, the method comprising the steps of:

- (a) creating a hibernaculum of a process containing some or all of the program modules, data and execution state of the active application,
- (b) sending the hibernaculum to a location for retrieval by the recipient; and
- (c) reconstructing the active application from the retrieved hibernaculum.

30

Preferably the location is the mailbox of the recipient or a personal information manager of the recipient and hibernaculum may be sent as an attachment to an e-mail sent to the recipient's mailbox. Prior to step (c) the hibernaculum may be transferred by the recipient to a further location and opened from the further location and the further
5 location may be a calendar or scheduler.

Preferably, the hibernaculum is sent to a storage location remote from the recipient and a reference to the storage location may be sent to a mailbox or a personal information manager of the recipient, the storage location being most preferably on a World-wide
10 Web server with the reference being a URL link. The reference may be transferred by the recipient to a further mailbox or personal information manager and opened from there.

According to the invention in a second aspect, there is provided a method of transferring
15 an active application from a sender to a recipient, one being an initiating party and the other being a target, the method comprising the steps of:

- (a) specifying the address of the target that is to participate in the transfer with the initiating party;
- (b) creating a hibernaculum of a process containing some or all of the program
20 modules, data and execution state of the active application;
- (c) sending the hibernaculum to the recipient;
- (d) reconstructing the active application from the hibernaculum.

Either the sender or the recipient may initiate the transfer.
25

"Sender" and "Recipient" are defined broadly. For example, the sender and recipient may be different machines, different users or the same entity spaced in time.

Apparatus for performing one or more of the methods described above is also envisaged
30 within the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described, by way of example, with
5 reference to the accompanying drawings, in which:-

Fig.1 is a general schematic model of an operating environment of a computing system,

Fig.2 schematically illustrates a process life-cycle and operations that may be performed on a process,

10 Fig.3 is a flow-chart illustrating the Hibernaculum Construct operation,

Fig.4 is a flow-chart illustrating the Assimilate operation, and

Fig.5 is a flow-chart illustrating the Mutate operation.

Fig.6 is a view of a computer screen illustrating the task manager of the embodiment of the invention together with an active application.

15 Fig.7 shows use on the Send command of the task manager illustrated in Fig. 6.

Fig.8 shows the computer screen of a recipient of the file sent to an E-mail application using the Send command of Fig. 7.

Fig. 9 shows the active application reconstructed on the recipient's machine.

Fig. 10 is a view similar to Fig. 8 showing the file sent to a Calendar application.

20

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiment of the invention uses process suspension and migration techniques discussed in the applicant's copending PCT applications PCT/SG98/00102
25 and PCT/SG99/00018, the contents of which are incorporated herein by reference. In order to provide background to the present invention, however, description of these techniques will first be given, after which the application of these techniques in the embodiment of the present invention will be described.

30 In this specification the following terms will be used with the following meaning:

"First class entity": an object that can be manipulated directly.

“Process”: a combination of data, program module(s) and current execution state.

“Execution state”: the values and contents of transient parameters such as the contents of registers, frames, counters, look-up tables and the like.

“Hibernaculum”: a data type used to store a suspended process.

5

Figure 1 shows the general model of a computing system. An application program 30 comprises data 10 and program modules 20. The operating system 60, also known as the virtual machine, executes the application 30 by carrying out the instructions in the program modules 20, which might cause the data 10 to be changed. The execution is
10 effected by controlling the hardware of the underlying machine 70. The status of the execution, together with the data and results that the operating system 60 maintains for the application 30, form its execution state 40.

Such a model is general to any computing system. It should be noted here that the present
15 invention starts from the realisation that all the information pertaining to the application at any time is completely captured by the data 10, program modules 20 and execution state 40, known collectively as the process 50 of the application 30.

The process 50 can have one or more threads of execution at the same time. Each thread
20 executes the code of a single program module at any given time. Associated with the thread is a current context frame, which includes the following components:

- A set of registers
- A program counter, which contains the address of the next instruction to be executed
- 25 • Local variables of the module
- Input and output parameters of the module
- Temporary results of the module

In any module A, the thread could encounter an instruction to invoke another module B.
30 In response, the program counter in the current frame is incremented, then a new context frame is created for the thread before it switches to executing module B. Upon

completing module B, the new context frame is discarded. Following that, the thread reverts to the previous frame, and resumes execution of the original module A at the instruction indicated by the program counter, i.e., the instruction immediately after the module invocation. Since module B could invoke another module, which in turn could invoke some other module and so on, the number of frames belonging to a thread may grow and reduce with module invocations and completions. However, the current frame of a thread at any given time is always the one that was created last. For this reason, the context frames of a thread are typically stored in a stack with new frames being pushed on and popped from the top. The context frames of a thread form its execution state, and the state of all the threads within the process 50 constitute its execution state 40 in Fig.1.

The data 10 and program modules 20 are shared among all threads. The data area is preferably implemented as a heap, though this is not essential. The locations of the data 10 and program modules 20 are summarized in a symbol table. Each entry in the table gives the name of a datum or a program module, its starting location in the address space, its size, and possibly other descriptors. Instead of having a single symbol table, each process may alternatively maintain two symbol tables, one for data alone and the other for program modules only, or the process could maintain no symbol table at all.

It is preferred that the data and program code of a process are stored in a heap and a program area respectively and are shared by all the threads within the process. In addition the execution state of the process comprises a stack for each thread, each stack holding context frames, in turn each frame containing the registers, local variables and temporary results of a program module, as well as addresses for further module invocations and returns. Before describing an embodiment of the invention in more detail, however, it is first necessary to introduce some definitions of data types and functions that are used in the embodiment and which will be referred to further below.

In addition to conventional data types such as integers and pointer, four new data types Data, Module, Stack and Hibernaculum are defined as follows:

Data: A variable of this data type holds a set of data references. Members are added to and removed from the set by means of the following functions;

Int AddDatum(Data d, String dataname) inserts the data item dataname in the heap of the process as a member of d.

5 Int DelDatum(data d, String dataname) removes the data item dataname from d.

Module: A variable of this data type holds a set of references to program modules. Members are added to and removed from the set with the following functions;

10 Int AddModule(Module d, String modulename) inserts the program module modulename in the program area of the process as a member of d.

Int DelModule(Module d, String modulename) removes the program module modulename from d.

15

Stack: A variable of this data type holds a list of ranges of execution frames from the stack of the threads. The list may contain frame ranges from multiple threads, however no thread can have more than one range. Variables of this type are manipulated by the following functions:

20

Int OpenFrame(Stack d, Thread threadname) inserts into d a new range for the thread threadname, beginning with the thread's current execution frame. This function has no effect if the thread already has a range in d.

25

Int CloseFrame(Stack d, Thread threadname) ends the open-ended range in d that belongs to the thread threadname. This function has no effect if the thread does not currently have an open-ended range in d.

30

Int PopRange(Stack d, Thread threadname) removes from d the range belonging to the thread threadname.

Hibernaculum: A variable of this data type is used to hold a suspended process.

As will be explained in more detail below a process may be suspended and stored in a hibernaculum prior to being transferred from one operating environment to another operating environment and/or may be subject to the following evolutionary operation:

Hibernaculum Construct(Stack s, Module m, Data d): This operation creates a new process with the execution state, program table and data heap specified as input parameters. The process is immediately suspended and then returned in a hibernaculum. The hibernaculum may be signed by the originating process as indication of its authenticity. Fig.3 is a flow-chart showing the hibernaculum construct operation.

A hibernaculum may be sent between operating environments by the following send and receive functions:

Int Send(Hibernaculum h, Target t) transmits the process contained within h to the specified target.

Hibernaculum Receive(Source s) receives from the specified source a hibernaculum containing a process.

A hibernaculum may be subject to the following evolutionary function:

Int Assimilate(Hibernaculum h, OverrideFlags f) activates the threads of the process stored within h and runs them as threads within a calling process's operating environment. Where there is a conflict between the data and/or program modules of the hibernaculum and the operating environment, the override flags specify which to preserve. Fig.4 is a flow-chart illustrating the steps of the assimilate operation.

Int Mutate(Stack s, int sflag, Module m, int mflag, Data d, int dflag) modifies the execution state, program table and data heap of the calling process. If a thread has an

entry in s, only the range of execution frames specified by this entry is preserved, the other frames are discarded. Execution stacks belonging to threads without an entry in s are left untouched. In addition, program modules listed in m and data items listed in d are kept or discarded depending on the flag status. Fig.5 is a flow-chart illustrating the steps of the mutate operation.

Fig.2 illustrates very schematically how these operations may act on a process 230 (which may be loaded from an application 210 or a hibernaculum 220). The process 230 may be subject to a Construct operation 110 to create a hibernaculum, a hibernaculum may be sent to a stream by a Send operation 120, or received from a stream by a Receive operation 130. The contents of a hibernaculum may be assimilated in the process by an Assimilate operation 140, and a process may be caused to mutate by a Mutate operation 150. The process 230 may of course also be subject to traditional operations.

An example will now be described in which it is assumed that an executing process p1 is required to migrate from a first host machine t1 to a second host machine t2.

To begin with the process p1 calls the following function:

Hibernaculum h = Construct(Stack s, Module m, Data d)

where s contains all the execution stacks in the process, m contains all the application specific modules in the process (ie m excludes those modules that are system specific to the first host machine), and d contains all the application specific data (ie d excludes data that are system specific to the first host machine). This function creates a new process p2 that contains only the data, program code and execution states of the original process p1 that are specific to the application and not to the system of the first host machine. Process p2 is immediately suspended and returned within a hibernaculum h.

The original process p1 then calls the function:

Int Send(hibernaculum h, Host t2)

which transmits the process p2 contained within hibernaculum h as a stream to the new host machine t2.

5

At the new host machine t2 a new process p3 is created containing initial system specific data and program code relating to the new host t2. This new process p3 then immediately executes the function:

10 Hibernaculum h = Receive(Host t1)

which receives from the first host machine t1 the hibernaculum h containing the process p2 which includes only the application specific data, program codes and execution states.

15 Process p3 then calls the function:

Int Assimilate(Hibernaculum h)

20 to activate the threads of the process p2 stored in h and to run them as new threads within the environment of the calling process p3. The original thread in p3 then terminates, resulting in a process that has all the application specific portions of process p1, but with the system specific portions replaced to suit the requirements of the second host machine t2.

25 The original process p1 terminates.

In the example described above the process p1 discards first host specific information prior to migration. However, the discarding may be done after migration to the second host. This is described in the following embodiment.

30

To begin the process p1 calls the following function:

Hibernaculum h = Construct(Stack s, Module m, Data d)

5 where s contains all the execution stacks in the process, m contains all modules in the process (including both system specific and application specific modules), and d contains all data in the process (including both system specific and application specific data). Thus the entire process p1 is contained within the hibernaculum as a new process p2 that is identical to p1. Process p2 is immediately suspended and returned within hibernaculum h.

10 The original process p1 then calls the function:

Int Send(hibernaculum h, Host t2)

15 which transmits the process p2 contained within the hibernaculum h as a stream to the new host machine t2 where the process p2 is then re-activated. The process p2 then calls the function:

Int Mutate(Stack s, int sflag, Module m, int mflag, Data d, int dflag)

20 where s contains all the execution stacks in the process, m contains all the application specific modules in the process, and d contains all the application specific data in the process, and where the flags are set to retain the contents of s, m and d. In this way all execution states, and all application specific modules and data are retained in the process p2, but all data and modules specific to the system of the first host are discarded. Process
25 p2 is then stored within a hibernaculum h in the second host using the construct operation.

At the new host machine t2 a new process p3 is created containing initial system specific data and program code relating to the new host t2. This process p3 then calls the function:

30

Int Assimilate(Hibernaculum h)

to activate the threads of the process p2 stored in h and to run them as new threads within the environment of the calling process p3. The original thread in p3 then terminates, resulting in a process that has all the application specific portions of p1, but with the system specific portions replaced to suit the requirements of the second host machine t2.

It will also be understood that in a variation of this second embodiment the mutate and assimilate functions may be reversed. That is to say, following the transfer of process p2 (identical to p1) from host t1 to host t2, process p2 may assimilate process p3 (containing the t2 system specific data and code) before the mutate operation is used to discard the t1 system specific data and code.

Thus it will be seen that there is provided a method by means of which a process can migrate from one host machine to another host machine and in doing so can adapt to the system requirements and configurations of the second host machine. Such a method allows processes to be readily transferred between host machines thus substantially facilitating the creation of a mobile computing environment.

To implement the process migration system in a Java environment, a package called snapshot is introduced. This package contains the following classes, each of which defines a data structure that is used in the migration and adaptation operations:

```
public class Hibernaculum {  
    ...  
}  
  
public class State {  
    ...  
}  
  
public class Module {
```

```
        ...  
    }  
  
    public class Data {  
5        ...  
    }  
  
    public class Machine {  
        ...  
10    }
```

In addition, the package contains a Snapshot class that defines the migration and adaptation operations:

```
15    public class Snapshot {  
        private static native void registerNatives();  
        static {  
            registerNatives();  
        }  
20  
        public static native Hibernaculum Construct(State s, Module m, Data d);  
        public static native int Send(Hibernaculum h, OutputStream o);  
        public static native Hibernaculum Receive(InputStream i);  
        public static native int Assimilate(Hibernaculum h, int f);  
25    public static native int Mutate(State s, int sflag, Module m, int mflag, Data d, int  
        dflag)  
  
        // This class is not to be instantiated  
        private Snapshot() {  
30    }  
    }
```

The methods in the Snapshot class can be invoked from application code. For example:

```

5      try {
          if (snapshot.Snapshot.Construct(s, m, d) != null) {
              // hibernaculum has been created
          } else {
              // failed to create hibernaculum
10      }
          catch(snapshot.SnapshotException e) {
              // Failed to create hibernaculum
          }

```

15 The migration and adaptation operations are implemented as native codes that are added to the Java virtual machine itself, using the Java Native Interface (JNI). To do that, a Java-to-native table is first defined:

```

#define KSH "Ljava/snapshot/Hibernaculum;"
20  #define KSS "Ljava/snapshot/State;"
    #define KSM "Ljava/snapshot/Module;"
    #define KSD "Ljava/snapshot/Data;"

    static JNINativeMethod snapshot_Snapshot_native_methods[] = {
25      {
          "Construct",
          ("("KSSKSMKSD")"KSH,
          (void*)Impl_Snapshot_Construct
      },
30      {
          "Send",

```

```

        (“KSH”Ljava/io/OutputStream;)I”,
        (void*)Impl_Snapshot_Send
    },
    {
5        “Receive”,
        (“Ljava/io/InputStream;)”KSH,
        (void*)Impl_Snapshot_Receive
    },
    {
10        “Assimilate”,
        (“KSH”)I”,
        (void*)Impl_Snapshot_Assimilate
    },
    {
15        “Mutate”
        (“KSSKSM”I”KSD”)I”
        (void*)Impl_Snapshot_Mutate
    },
};

```

After that, the native implementations are registered via the following function:

[illegible]

Besides the above native codes, several functions are added to the Java virtual machine implementation, each of which realizes one of the migration and adaptation operations:

```
void* Impl_Snapshot_Construct(..) {  
5      // follow flowchart in Figure 3  
  
      ...  
}  
  
10 void* Impl_Snapshot_Send(..) {  
      // send given hibernaculum to specified target  
      ...  
}  
  
15 void* Impl_Snapshot_Receive(..) {  
      // receive a hibernaculum from a specified source  
      ...  
}  
  
20 void* Impl_Snapshot_Assimilate(..) {  
      // follow flowchart in Figure 4  
      ...  
}  
  
25 void* Impl_Snapshot_Mutate(..){  
      //follow flowchart in Figure 5  
      ...  
}
```

30 The embodiment of the present invention is concerned with utilising the techniques described above in a manner that can be easily performed by the user.

In this respect, an application termed a task manager providing a visual control panel on the user's desktop, is provided to perform the process suspension and migration operations and functions. The computing processes are active Java applications running on a Microsoft Windows machine. To suspend and resume these processes, the Java
5 virtual machine is enhanced as described above. This enhanced JVM, installed on every user machine, can start up new Java applications and resume suspended processes.

When starting or resuming a process, the JVM activates the task manager if it is not
10 already running on the user machine. Once started, the task manager listens at a pre-defined port. Each new process contacts the task manager at that port, and the two then establish a two-way socket connection between them. Using the connection, the process updates the task manager on any changes in status, including process termination, window creation and closure. The connection is also used by the task manager to send
15 commands to the process.

The task manager 110 shown on the computer screen desktop 100 of a user is illustrated in Fig. 6. Every computer process on the user's machine is registered with the task manager and in selection box 115 the name of the window(s) owned by the process and
20 other information such as status etc. are listed. In Fig. 6, a bar chart process, owning the window 200 with the label "Test 29", is shown open on the desktop.

The task manager has function buttons 120 – 127 as follows:

25 E-mail button 120: Clicking on this button allows the user to suspend a process and send it to a recipient as an attachment to an e-mail message.

Save button 121: Clicking on this button allows the user to suspend a process and deposit it in a database. An e-mail containing a reference to the location of the process
30 may optionally be sent to the intended recipient, so that the process can be retrieved later by clicking on the reference.

Open button 122: Clicking on this button allows the user to retrieve a saved process. A pop-up list of process that the user is allowed to retrieve is shown. On selection of the process, this is reconstructed on the user's machine.

5

Push button 123: Clicking on this button allows the user to suspend a process, then transfer (move or copy) it directly to another machine. The process is selected first and the destination address (target machine name and login information) is entered to activate the transfer.

10

Pull button 124: Clicking on this button allows the user to transfer (move or copy) a process from another machine. Upon entry of the source address (target machine name and login information), a pop-up list of active processes that the user can transfer is shown. Selection of the desired process causes the process to be transferred.

15

New button 125: Clicking on this button allows the user to select an application and run it as a new process on the machine.

Exit 126 and Info (help) 127 buttons have their usual meanings.

20

The functions identified by function buttons 120-124 are performed as follows:

1) E-mail Function

25

- a) The sender picks process to send and clicks on the e-mail button 120 (Fig. 6);
- b) A hibernaculum of the process is created using the Hibernaculum Construct operation;
- c) An output stream is opened to a target in the form of a file and the hibernaculum is transferred to the target file using the Int Send function;
- d) An e-mail sending program (e.g. Microsoft Outlook) is called for input of the destination address of the recipient, subject and any text (Fig. 7);

30

- e) The file is attached to the e-mail and sent; (the e-mail function ends here, the next steps being performed by the recipient)
- f) The recipient on his machine opens the e-mail 135 received in his mailbox using an e-mail application (e.g. Microsoft Outlook). and selects the file attachment 140 in the received e-mail (Fig. 8)
- 5 g) This causes the suspended process in the file attachment 140 to be reconstructed. Since the task manager is not already running, it is activated. After that, the reconstructed process registers itself with the task manager (Fig. 9).
- 10 h) The reconstructed process opens an input stream from the (source) file attachment 140 and the hibernaculum is read from the input stream using the Hibernaculum Receive function
- i) The reconstructed process absorbs the hibernaculum using the Int Assimilate function (Fig. 9), thereby resuming the suspended process 200.

15

The screens presented to the user (sender or recipient, where appropriate) at certain stages in the performance of steps (a)-(h) are illustrated in Figs. 6-9 as noted. Similar screens (not shown) adapted to the tasks to be performed are provided for the other functions.

20 2) Save Function

- a) The sender picks process to send and clicks on the Save button 121;
- b) A hibernaculum of the process is created using the Hibernaculum Construct operation;
- c) An output stream is opened to a target in the form of a file and the hibernaculum is transferred to the target file using the Int Send function;
- 25 d) A file-sending program is called for entry of the address of the save location. The save location is on a web server running at a pre-specific machine and port;
- e) The file is sent to the location;

- f) An e-mail sending program is called for input of the destination address of the recipient and subject. The body of the e-mail contains any text and a handle (a URL link) specifying the location of the file;
- g) The e-mail is sent to the recipient; (the Save function ends here, the next steps being performed by the recipient)
- h) The recipient on his machine clicks on the handle in the received e-mail
- i) This causes the suspended process associated with the handle to be reconstructed. If the task manager is not already running, it is activated. Following this, the reconstructed process registers itself with the task manager.
- j) The reconstructed process opens an input stream from the (source) file specified by the handle and the hibernaculum is read from the input stream using the Hibernaculum Receive function.
- k) The reconstructed process absorbs the hibernaculum using the Int Assimilate function, thereby resuming the suspended process.

In order to open the e-mail attachment or server-based file, a new MIME-type is registered on the recipient's machine so that, when the recipient clicks on file attachment or URL link, the file is automatically opened with the enhanced JVM. On Windows 95/98, this is done by double-clicking the "My Computer" icon, picking the "Options" entry in the "View" menu, clicking the "File Types" tab, and finally adding a "New Type" for "application/x-java-task" with the ".jpm" extension that is associated with the enhanced-JVM program.

The user may transfer (move or copy) the file attachment or URL link from its receipt location (e-mail In box) to another location or application. One particularly preferred transfer is to a calendar 300 or scheduler in which the attachment/link 310 is further associated with scheduling information or a deadline 320. The result of this transfer is illustrated in Fig. 10.

3) Open Function

- a) The user enters his login information, after which he is shown a list of handles of suspended processes (created through steps a)-e) of the Save function) on the web server;
- 5 b) Upon selecting a handle, steps (i)-(k) of the Save function are followed.

4) Push Function

- a) The sender picks the process to send from the list displayed in the task manager and clicks on the Push button 123;
- 10 b) The task manager notifies the recipient machine to create a new process. The task manager is activated if it is not already running.
- c) The sender machine opens an output stream to the new process on the recipient machine;
- d) The new process on the recipient machine calls the Hibernaculum Receive
15 function to operate on the (input) stream from the sender;
- e) The sender machine creates a hibernaculum of the process using the Hibernaculum Construct operation;
- f) The sender machine writes the hibernaculum to the output stream using the Int Send operation;
- 20 g) The recipient machine reconstructs the process from the hibernaculum using the Int Assimilate function.

5) Pull Function

- a) The recipient clicks on the Pull button 124, specifies the sender machine
25 and selects the process to be transferred;
- b) A new process is created on the recipient machine. The task manager is activated if it is not already running;
- c) The sender machine opens an output stream to the new process on the recipient machine;
- 30 d) The new process on the recipient machine calls the Hibernaculum Receive function to operate on the (input) steam from the sender;

- e) The sender machine creates a hibernaculum of the process using the Hibernaculum Construct operation;
- f) The sender machine writes the hibernaculum to the output stream using the Int Send operation;
- 5 g) The recipient reconstructs the process from the hibernaculum using the Int Assimilate function.

This embodiment is not to be construed as limitative. For example, the E-mail function sends the hibernaculum by e-mail to the mailbox of the recipient which may be accessed
10 by any suitable means, for example an e-mail application or personal information manager (PIM) used in any operating system. The transfer may be via any transfer medium, for example via Internet, Intranet, local area network or other communications link. The recipient may open his mailbox on the same machine as the sender, so that the transfer is within the machine. The Save function in a similar way can use any transfer
15 medium and deposit the hibernaculum in any storage location, for example on a server, or on the same or a different machine to the sender. Similar variations are envisaged for the Open, Push and Pull functions. The sender and recipient may be identified as different machines, different users, accounts or logins on the same or different machines or can be effectively the same, spaced simply in time, with the sender effectively saving the
20 process for subsequent reactivation by himself on the same machine later.

CLAIMS

1. A method of transferring an active application from a sender to a recipient, the method comprising the steps of:
 - 5 (d) creating a hibernaculum of a process containing some or all of the program modules, data and execution state of the active application,
 - (e) sending the hibernaculum to a location for retrieval by the recipient; and
 - (f) reconstructing the active application from the retrieved hibernaculum.
- 10 2. A method as claimed in claim 1 wherein the location is the mailbox of the recipient.
3. A method as claimed in claim 2 wherein the hibernaculum is sent as an attachment to an e-mail sent to the recipient's mailbox.
- 15 4. A method as claimed in claim 1, wherein the hibernaculum is sent to a personal information manager of the recipient.
5. A method as claimed in claim 1 wherein the hibernaculum is sent to a storage location remote from the recipient.
- 20 6. A method as claimed in claim 4 wherein a reference to the storage location is sent to a mailbox of the recipient.
- 25 7. A method as claimed in claim 5 wherein the storage location is on a World-wide Web server and the reference is a URL link.
8. A method as claimed in claim 1 wherein the hibernaculum is sent to a storage location remote from the recipient and a reference to the storage location is sent to a personal information manager of the recipient.
- 30

9. A method as claimed in any one of the preceding claims wherein, prior to step (c) the hibernaculum is transferred by the recipient to a further location and opened from the further location.
- 5 10. A method as claimed in claim 9 wherein the further location is a calendar or scheduler.
11. A method as claimed in any one of claims 6 to 8 wherein, prior to step (c) the reference is transferred by the recipient to a further mailbox or personal
10 information manager and opened from there.
12. A method of transferring an active application from a sender to a recipient, one being an initiating party and the other being a target, the method comprising the steps of:
- 15 (a) specifying the address of the target that is to participate in the transfer with the initiating party;
- (b) creating a hibernaculum of a process containing some or all of the program modules, data and execution state of the active application;
- (c) sending the hibernaculum to the recipient;
- 20 (d) reconstructing the active application from the hibernaculum.
13. A method as claimed in claim 12 wherein the sender initiates the transfer.
14. A method as claimed in claim 12 wherein the recipient initiates the transfer.
- 25 15. A method as claimed in any one of the preceding claims wherein the sender and recipient are different machines.
16. A method as claimed in any one of the preceding claims wherein the sender and
30 recipient are different users.

17. A method as claimed in any one of claims 1 to 14 wherein the sender and recipient are the same, the steps of sending and recovering being spaced in time.
18. Apparatus for performing the method of any one of the preceding claims.

5

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204

1/10

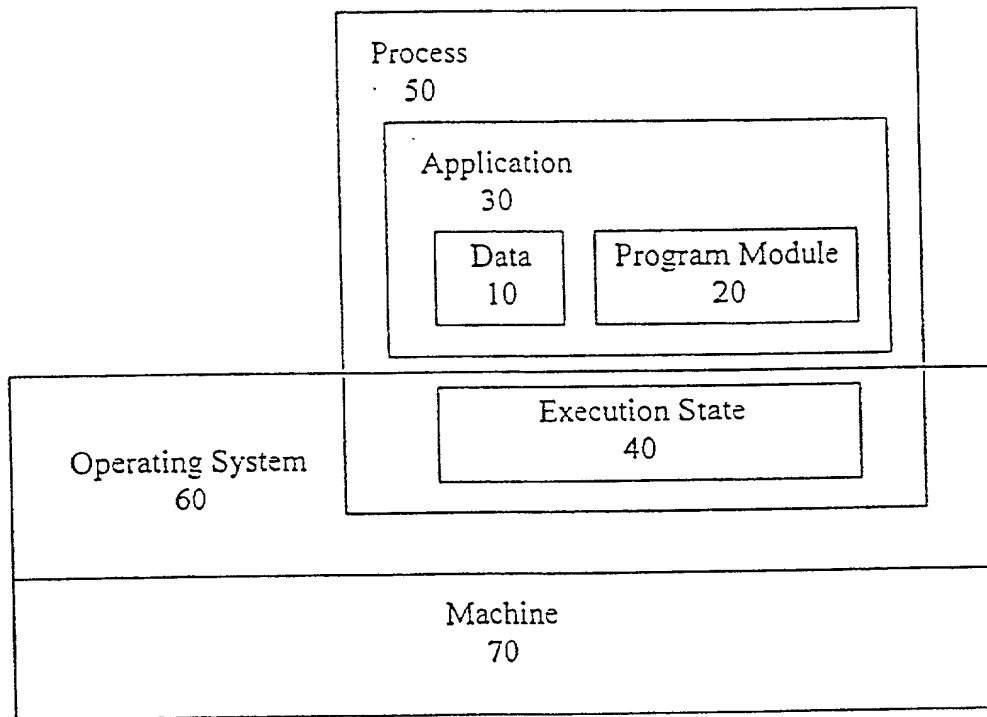


Fig.1

2/10

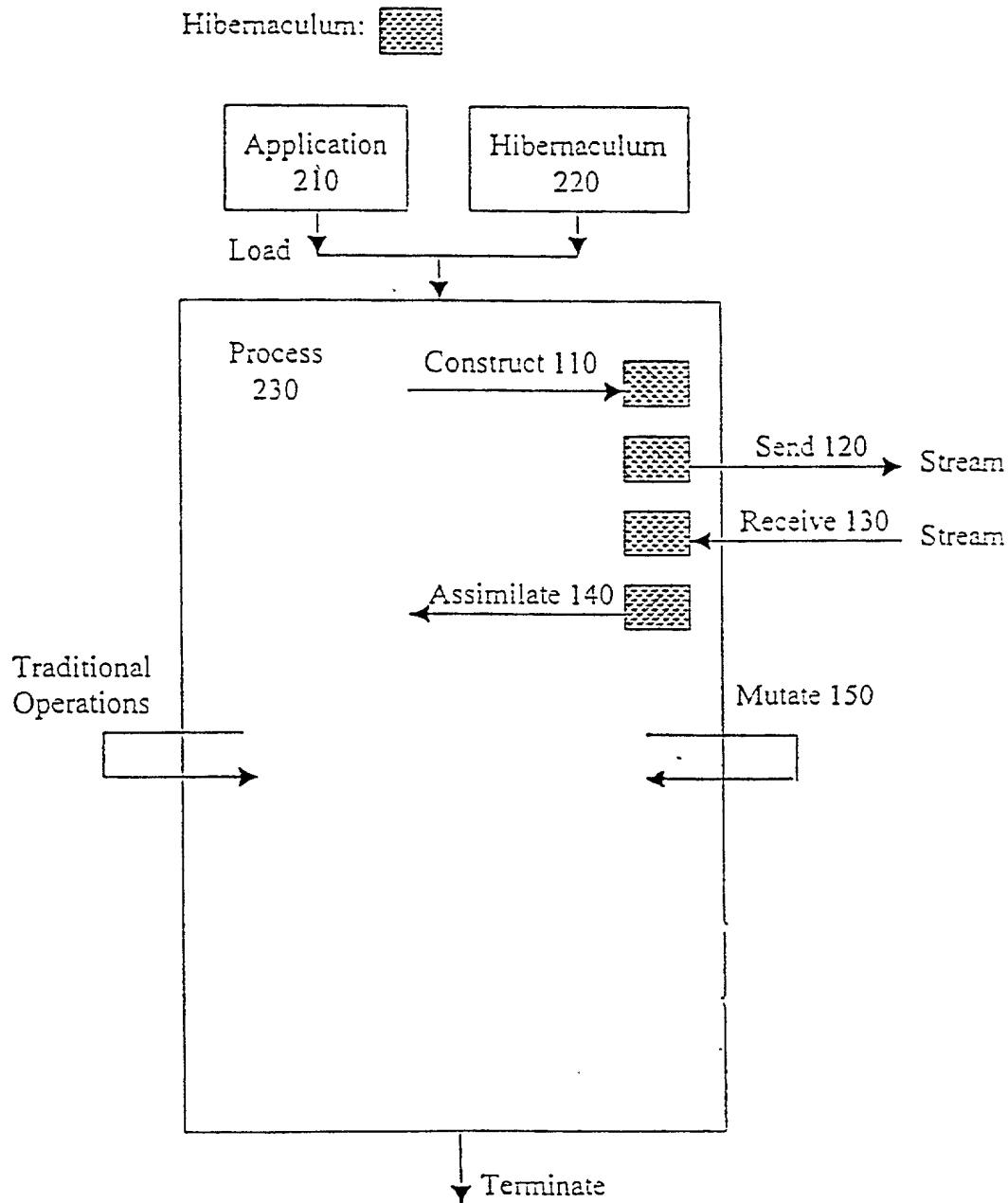


Fig.2

3/10

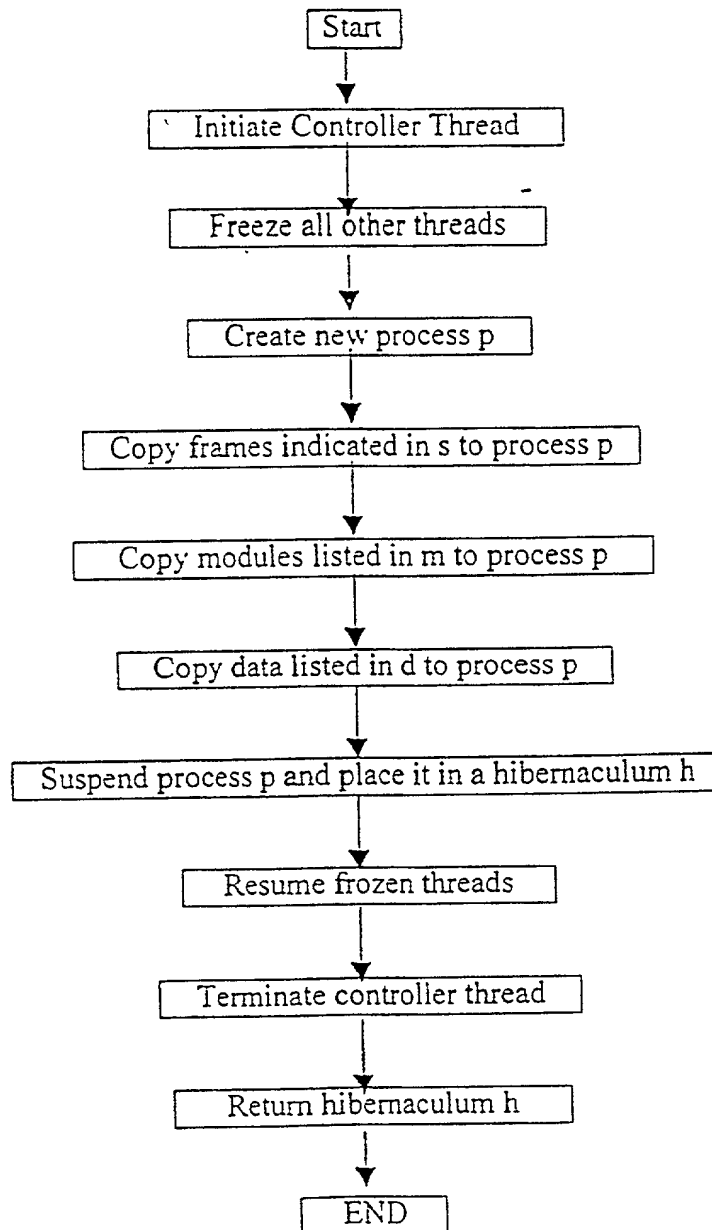


Fig.3

4/10

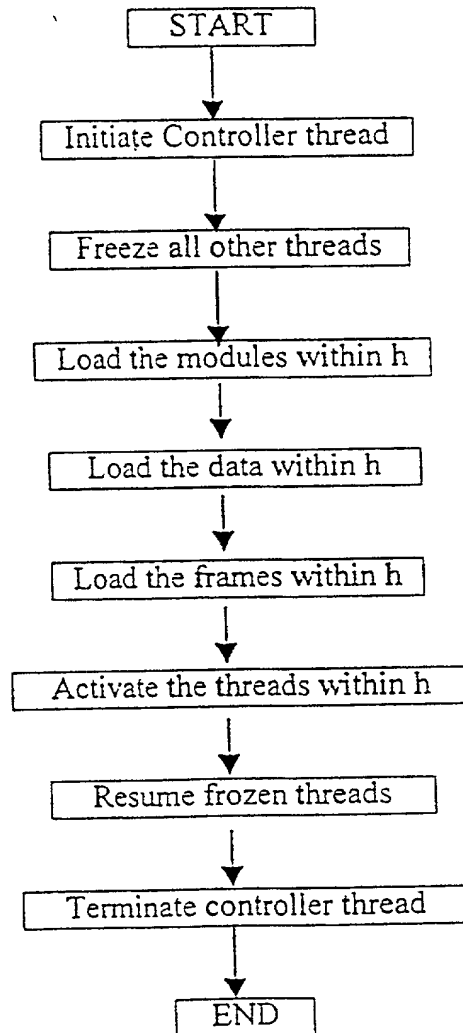


Fig.4

09/856514

5/10

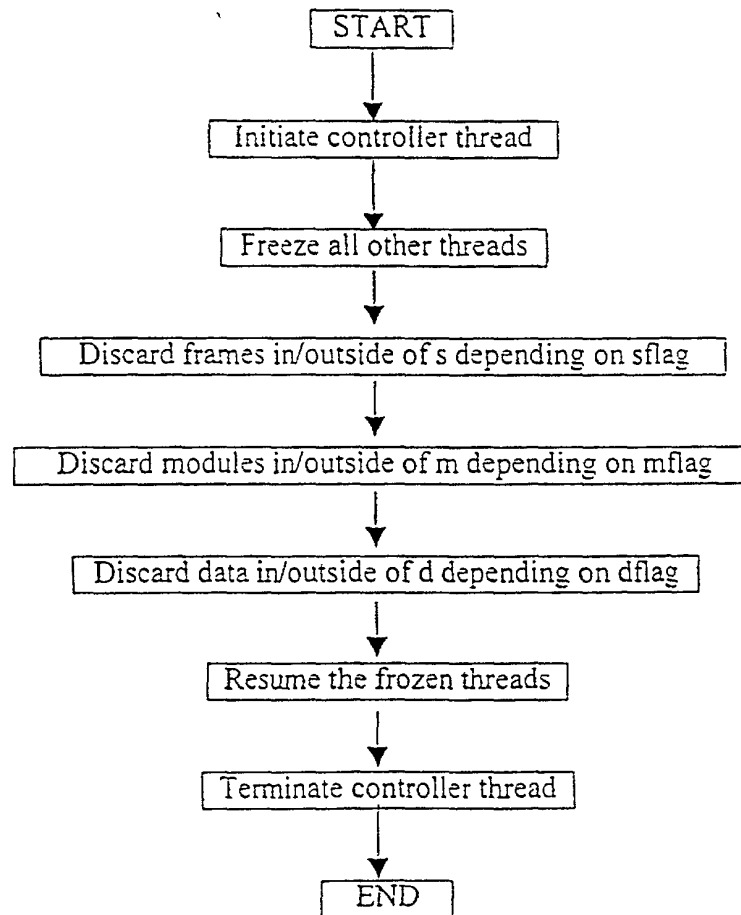


Fig.5

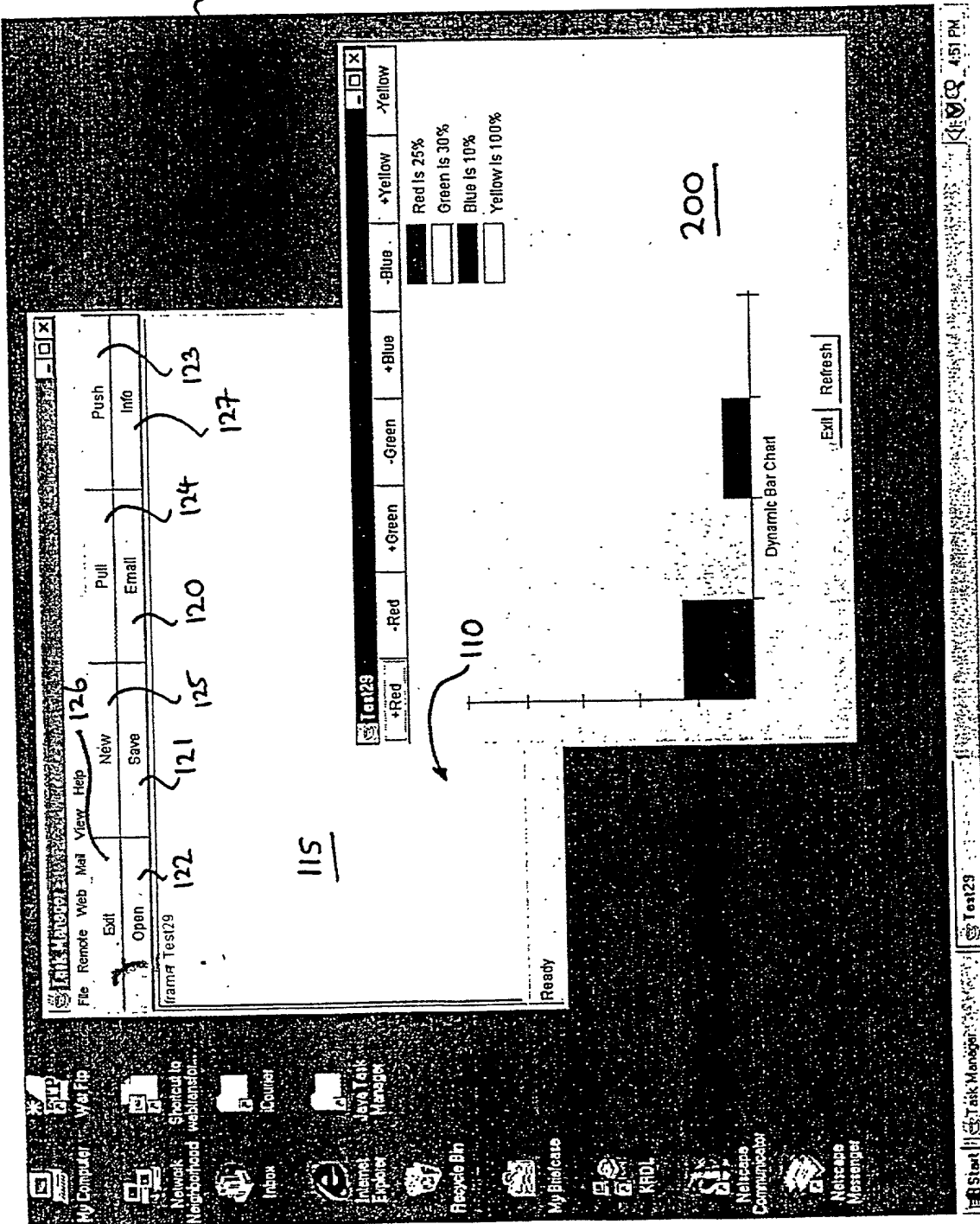


Fig. 6

09/856514

7/10

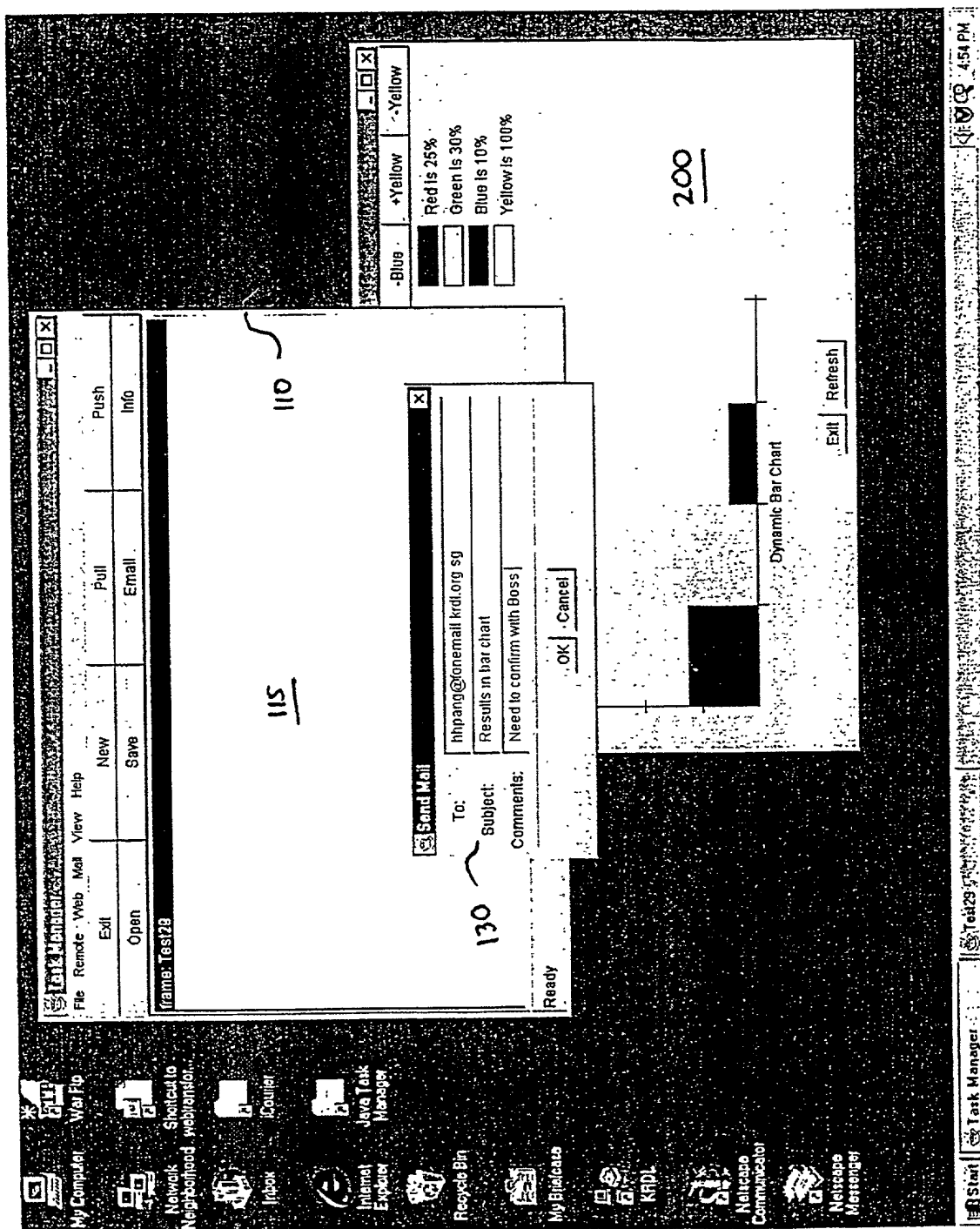


Fig. 7

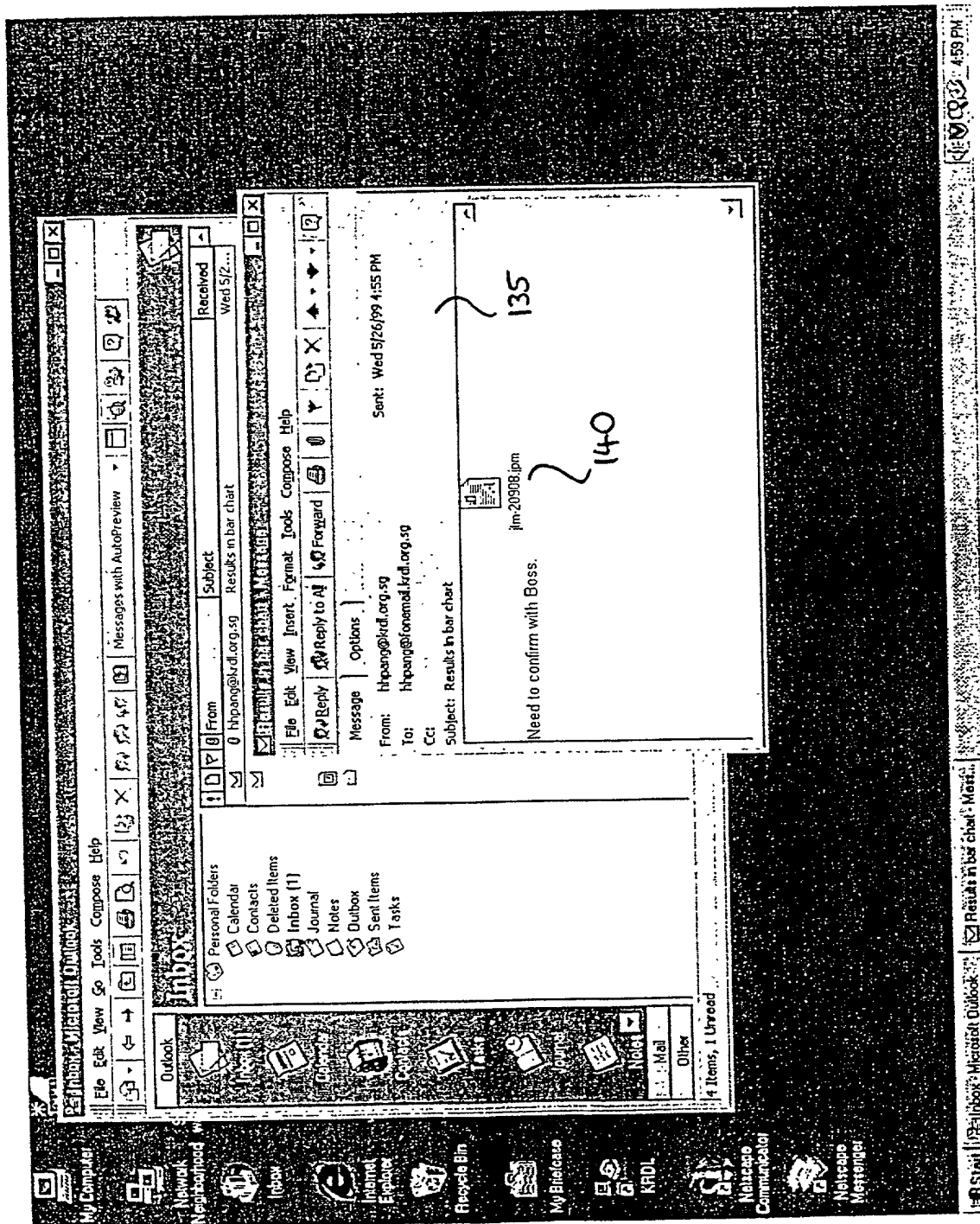


Fig. 8

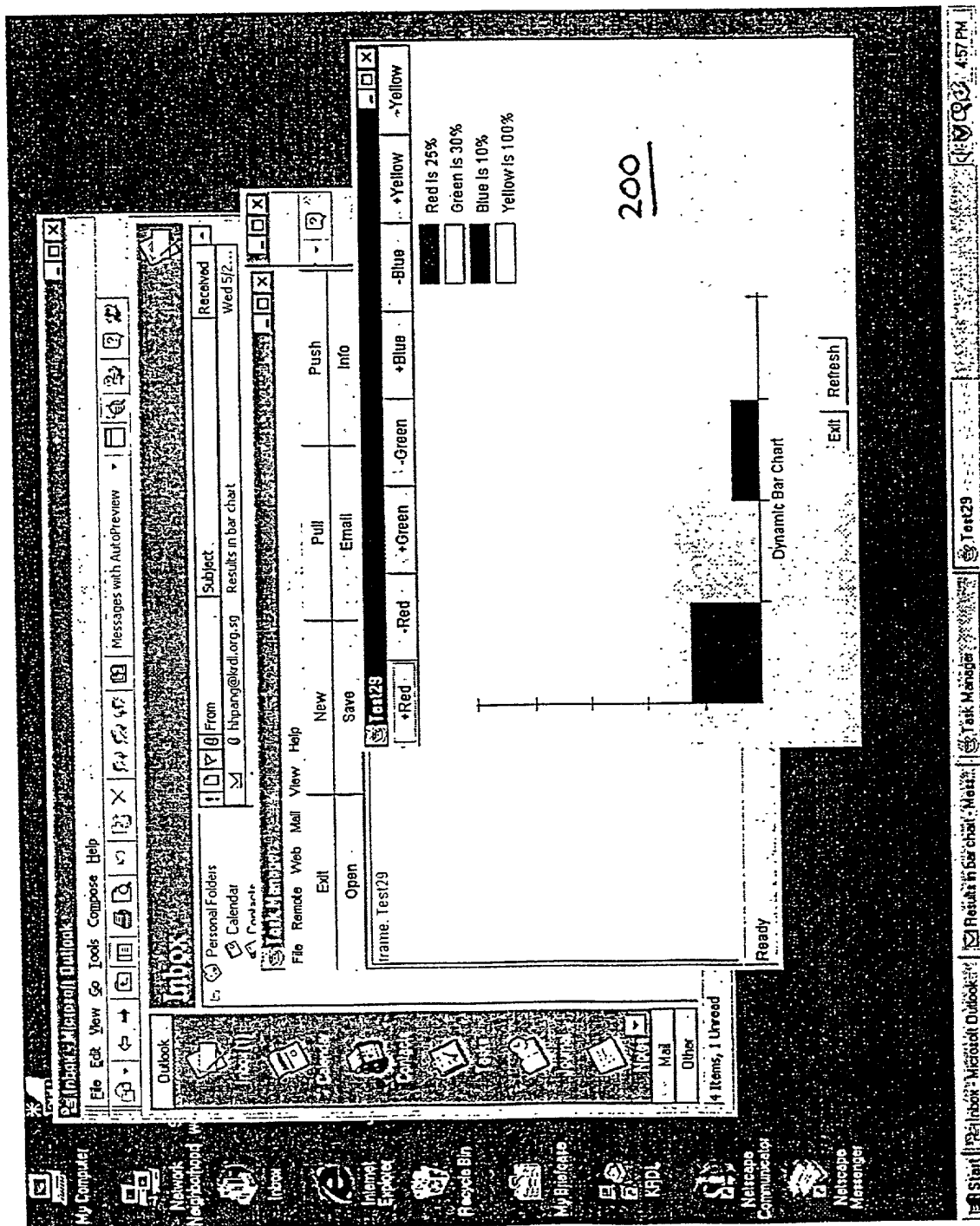


Fig. 9

09/856514

10/10

300

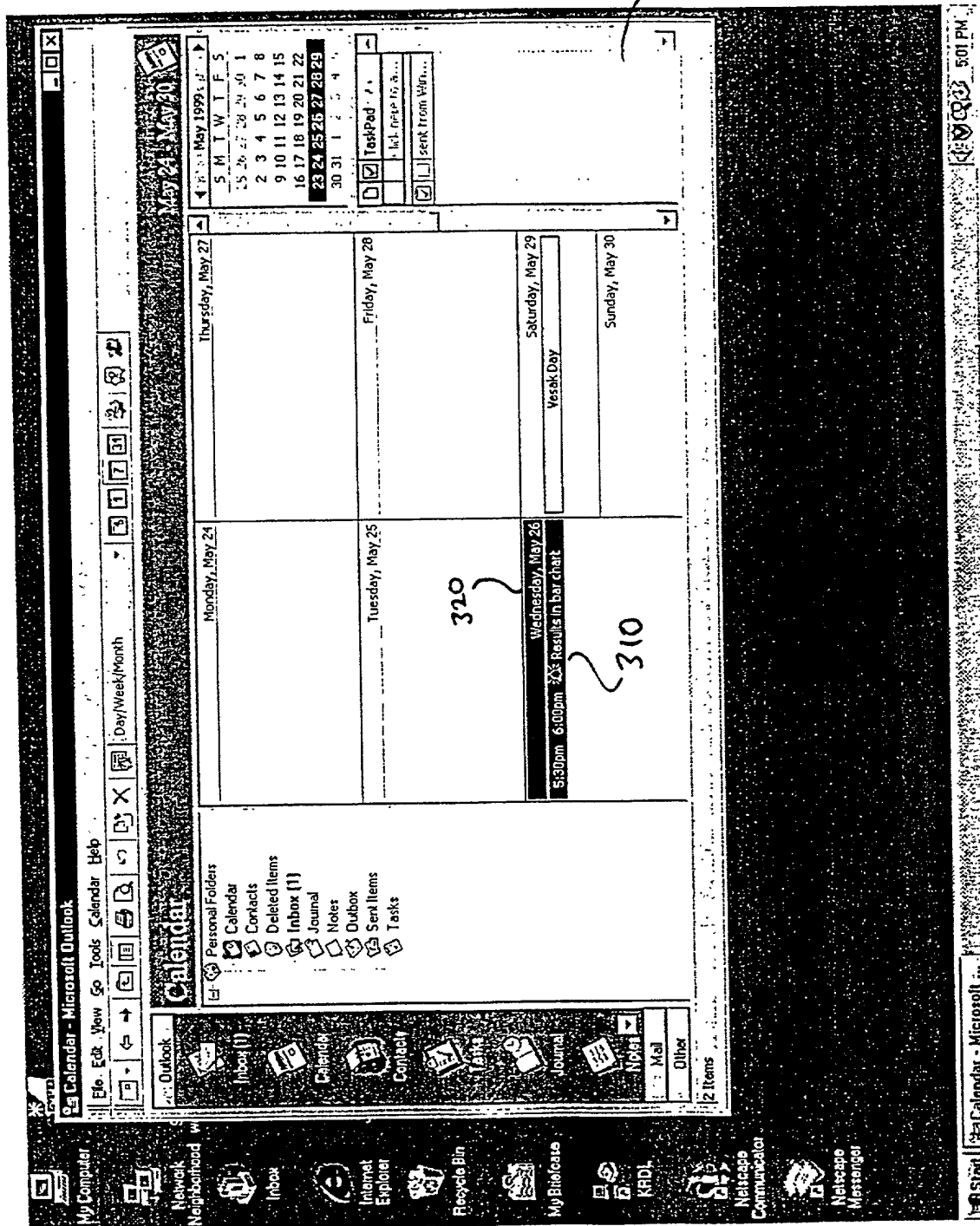


Fig. 10

Declaration and Power of Attorney For Utility or Design Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

A METHOD OF TRANSFERRING AN ACTIVE APPLICATION FROM A SENDER TO A RECIPIENT

the specification of which is attached hereto unless the following box is checked:

☒ was filed on 15 July 1999 as

United States Application Number _____ (if applicable) or,
and was amended on _____

PCT International Application Number PCT/SG99/00077 _____ (if applicable)
and was amended on _____

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code §119 (a-d) or §365(b) of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT international application which designated at least one country other than the United States of America, listed below. I have also identified below, by checking the "No" box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed:

			Priority Claimed	
PCT/SG98/00102	WO	16 DECEMBER 1998	<input checked="" type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
PCT/SG99/00018	WO	18 MARCH 1999	<input checked="" type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

☐ Additional foreign application numbers are listed on a supplemental priority sheet attached hereto.

I hereby claim the benefit under Title 35, United States Code §119(e) of any United States provisional application(s) listed below.

_____	_____
(Number)	(Day/Month/Year Filed)
_____	_____
(Number)	(Day/Month/Year Filed)
_____	_____
(Number)	(Day/Month/Year Filed)

☐ Additional provisional application numbers are listed on a supplemental priority sheet attached hereto.

I hereby claim the benefit under Title 35, United States Code §120 of any United States application(s), or §365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code §112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations §1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

(Application No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

☐ Additional U.S. or international application numbers are listed on a supplemental priority sheet attached hereto.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

The undersigned hereby authorizes the U.S. attorney or agent named herein to accept and follow instructions from either his foreign patent agent or corporate representative, if any, as to any action to be taken in the Patent and Trademark Office regarding this application without direct communication between the U.S. attorney or agent and the undersigned. In the event of a change in the persons from whom instructions may be taken, the U.S. attorney or agent named herein will be so notified by the undersigned.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the attorney(s) and/or agent(s) associated with the Customer Number provided below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith, and direct that all correspondence be addressed to that Customer Number:

CUSTOMER NUMBER 7055

The appointed attorneys include:

Neil F. Greenblum	Reg. No. 28,394	Stephen M. Roylance	Reg. No. 31,296
Bruce H. Bernstein	Reg. No. 29,027	Leslie J. Paperner	Reg. No. 33,329
Arnold Turk	Reg. No. 33,094	William Pieprz	Reg. No. 33,630
James L. Rowland	Reg. No. 32,674	William E. Lyddane	Reg. No. 41,568

At: Greenblum & Bernstein, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191

Direct Telephone Calls to: Greenblum & Bernstein, P.L.C. (703) 716-1191

Full name of sole or first inventor PANG Hwee Hwa (Family name PANG)

Inventor's signature

Date

11 June 2001

Residence

Singapore

Citizenship

Singapore

Post Office Address

201 Tanjong Rhu Road #15-11, Singapore 439617

(Supply similar information and signature for second and subsequent joint inventors.)

Full name of second joint inventor, if any		<u>GUO Bin</u>	(Family name Guo)
Second Inventor's signature		<i>Guo Bin</i>	Date <i>July 30, 2001</i>
Residence	<u>Singapore</u> <u>SSGIX</u>		
Citizenship	China		
Post Office Address	Blk 519 West Coast Road #11-617, Singapore 120519		
Full name of third joint inventor, if any			
Third Inventor's signature		Date	
Residence			
Citizenship			
Post Office Address			
Full name of fourth joint inventor, if any			
Fourth Inventor's signature		Date	
Residence			
Citizenship			
Post Office Address			
Full name of fifth joint inventor, if any			
Fifth Inventor's signature		Date	
Residence			
Citizenship			
Post Office Address			